



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/808,429

03/25/2004

Shyh-Kwei Chen

YOR920040052US1

6945

48150

7590

10/06/2010

MCGINN INTELLECTUAL PROPERTY LAW GROUP, PLLC

8321 OLD COURTHOUSE ROAD

SUITE 200

VIENNA, VA 22182-3817

EXAMINER

JOHNSON, JOHNESE T

ART UNIT

PAPER NUMBER

2166

MAIL DATE

DELIVERY MODE

10/06/2010

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/808,429	Applicant(s) CHEN ET AL.	
	Examiner Johnese Johnson	Art Unit 2166	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09 July 2010.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1 and 3-41 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1, 3-41 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Remarks

1. In response to the amendment filed on 7-9-2010, claims 1 and 3-41 are pending in this application.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1, and 3-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mitchell et al. (US PG Pub. No. 2005/0229186) in view of Kaipa et al (US PG Pub. No. 2005/0114394).

As to claims 1 and 25, Mitchell discloses:

receiving an object and a collaboration code (see paragraphs 33 and 35; wherein collaboration code is interpreted as procedure(s) to manipulate the data/ object);
determining a business object definition for said object based upon said collaboration code (see paragraphs 34, lines 4-8 and 38, lines 3-5; wherein collaboration code is interpreted as procedure(s) to manipulate the data/ object); and

However, Mitchell does not explicitly disclose:

Art Unit: 2166

storing said business object definition, wherein said collaboration code determines said business object definition for said object without pre-defined business object definitions, if the object does not conform to a known business object definition.

Kaipa discloses:

storing said business object definition, wherein said collaboration code determines said business object definition for said object without pre-defined business object definitions, if the object does not conform to a known business object definition (see paragraph 39, lines 1-3 and 7-10; wherein the definition stored in a business object (as in applicant's publication, paragraph 65) is interpreted as unknown because the schema is xml and the business object definition created is java).

It would have been obvious, at the time the invention was made, to have modified the teaching of Mitchell by the teaching of Kaipa to provide a better way to carry out the conversion of XML schema to Java and other object definitions (see Kaipa, paragraph 9).

As to claim 3, Mitchell as modified by Kaipa, discloses:

wherein said object comprises a business object (see Kaipa, paragraph 19), and wherein said determining the business object definition for said business object comprises reverse engineering said business object to examine how the business object was obtained (see Mitchell, paragraph 22).

As to claim 4, Mitchell as modified by Kaipa, discloses:

Art Unit: 2166

further comprising forwarding said object and said object definition (see Kaipa, paragraph 37, lines 2-6).

As to claim 5, Mitchell as modified by Kaipa, discloses:

wherein said forwarding comprises forwarding said object and said object definition to an application adapter (see Kaipa, paragraph 37, lines 2-6).

As to claim 6, Mitchell as modified by Kaipa, discloses:

further comprising processing said object based upon said object definition in said application adapter (see Kaipa, paragraph 37).

As to claim 7, Mitchell as modified by Kaipa, discloses:

wherein said collaboration code determines how data from a second object is mapped to said object (see Kaipa, paragraphs 18 and 35).

As to claim 8, Mitchell as modified by Kaipa, discloses:

wherein said collaboration code determines how said object is derived from said second object (see Kaipa, paragraphs 18 and 35).

As to claim 9, Mitchell as modified by Kaipa, discloses:

Art Unit: 2166

wherein said collaboration code determines how said object is derived from said second object and a second object definition (see Mitchell, paragraphs 22 and 25, lines 1-4).

As to claim 10, Mitchell as modified by Kaipa, discloses:

further comprising receiving said second object definition (see Mitchell, paragraph 33).

As to claim 11, Mitchell as modified by Kaipa, discloses:

wherein said determining comprises determining said object definition for said object based upon said collaboration code and said second object definition (see Mitchell, paragraphs 22 and 25, lines 1-4).

As to claim 12, Mitchell as modified by Kaipa, discloses:

wherein said receiving comprises receiving said object and said collaboration code from a broker (see paragraphs 33 and 35; wherein the broker is interpreted as dealing with input and output requests as in applicant's publication, paragraph 41, last line).

As to claim 13, Mitchell discloses:

a processor (see figs. 1-3; wherein the system inherently comprises a computer) for receiving an object and a collaboration code, and for determining an object definition for said object based upon said collaboration code (see paragraphs 33 and 35; wherein collaboration code is interpreted as procedure(s) to manipulate the data/ object),

However, Mitchell does not explicitly disclose:

Art Unit: 2166

wherein said collaboration code determines said business object definition for said object without pre-defined business object definitions, if the object does not conform to a known business object definition.

Kaipa discloses:

wherein said collaboration code determines said business object definition for said object without pre-defined business object definitions, if the object does not conform to a known business object definition (see paragraphs 34, lines 4-8 and 38, lines 3-5; wherein collaboration code is interpreted as procedure(s) to manipulate the data/object).

It would have been obvious, at the time the invention was made, to have modified the teaching of Mitchell by the teaching of Kaipa to provide a better way to carry out the conversion of XML schema to Java and other object definitions (see Kaipa, paragraph 9).

As to claim 14, Mitchell as modified by Kaipa, discloses:

wherein said object comprises a business object (see Kaipa, paragraph 19),

wherein said processor comprises a reverse object discovery agent (see Mitchell, paragraph 22), and

wherein said reverse object discovery agent conducts said determining the business object definition for said business object by reverse engineering said business object to examine how the business object was obtained (see Mitchell, paragraph 22).

As to claim 15, Mitchell as modified by Kaipa, discloses:
further comprising means for forwarding said object and said object definition to an application adapter (see Kaipa, paragraph 37, lines 2-6).

As to claim 16, Mitchell as modified by Kaipa, discloses:
wherein said collaboration code determines how data from a second object is mapped to said object (see Kaipa, paragraphs 18 and 35).

As to claim 17, Mitchell as modified by Kaipa, discloses:
further comprising means for receiving a second object definition, wherein said collaboration code determines how said object is derived from said second object and said second object definition (see Mitchell, paragraphs 22 and 25, lines 1-4).

As to claim 18, Mitchell as modified by Kaipa, discloses:
wherein said means for determining comprises means for determining said object definition for said object based upon said collaboration code and said second object definition (see Mitchell, paragraphs 22 and 25, lines 1-4).

As to claim 19, Mitchell as modified by Kaipa, discloses:
wherein said means for receiving comprises means for receiving said object and said collaboration code from a broker (see paragraphs 33 and 35; wherein the broker is

Art Unit: 2166

interpreted as dealing with input and output requests as in applicant's publication, paragraph 41, last line).

As to claim 26, Mitchell as modified by Kaipa, discloses:

further comprising analyzing said collaboration code and said input object definition (see Mitchell, paragraph 22 – reverse engineering).

As to claim 27, Mitchell as modified by Kaipa, discloses:

further comprising creating a new object definition based upon the results of said analyzing (see Mitchell, paragraph 22, lines 1-7).

As to claim 28, Mitchell as modified by Kaipa, discloses:

further comprising forwarding said object if said object conforms to a known object definition (see Mitchell, paragraph 34, lines 4-8).

As to claim 29, Mitchell as modified by Kaipa, discloses:

wherein said object comprises a business object (see Kaipa, paragraph 19).

As to claim 30, Mitchell as modified by Kaipa, discloses:

Art Unit: 2166

further comprising forwarding said new object definition to an application adapter (see Kaipa, paragraph 37, lines 2-6).

As to claim 31, Mitchell as modified by Kaipa, discloses: further comprising receiving a subscription from said application adapter for said new object definition (see Kaipa, paragraph 37; wherein "subscription from" is interpreted as receiving the new object definition).

As to claim 32, Mitchell as modified by Kaipa, discloses:

further comprising forwarding said object in response to said subscription (see Kaipa, paragraph 39, "The resultant business object definition (in this case, preferably a Java object) is forwarded to the connector 310 for use in runtime object conversion").

As to claim 33, Mitchell discloses:

integrating computer-readable code into a computing system (see paragraph 18),

the computer-readable code comprising:

instructions for receiving an object and a collaboration code (see paragraphs 33 and 35; wherein collaboration code is interpreted as procedure(s) to manipulate the data/object);

instructions for determining a business object definition for said object based upon said collaboration code (see paragraphs 34, lines 4-8 and 38, lines 3-5; wherein

Art Unit: 2166

collaboration code is interpreted as procedure(s) to manipulate the data/ object); and

However, Mitchell does not explicitly disclose:

instructions for storing said business object definition,

wherein said collaboration code determines said business object definition for said

object without pre-defined business object definitions, if the object does not conform to

a known business object definition.

Kaipa discloses:

instructions for storing said business object definition (see paragraph 34, lines 18-19),

wherein said collaboration code determines said business object definition for said

object without pre-defined business object definitions, if the object does not conform to

a known business object definition (see paragraph 39, lines 1-3 and 7-10; wherein the

definition stored in a business object (as in applicant's publication, paragraph 65) is

interpreted as unknown because the schema is xml and the business object definition

created is java).

It would have been obvious, at the time the invention was made, to have modified the teaching of Mitchell by the teaching of Kaipa to provide a better way to carry out the conversion of XML schema to Java and other object definitions (see Kaipa, paragraph 9).

As to claim 34, Mitchell as modified by Kaipa, discloses:

wherein said object comprises a business object (see Kaipa, paragraph 19).

Art Unit: 2166

As to claim 35, Mitchell as modified by Kaipa, discloses:

further comprising instructions for forwarding said new object definition to an application adapter (see Kaipa, paragraph 37, lines 2-6).

As to claim 36, Mitchell as modified by Kaipa, discloses:

further comprising instructions for receiving a subscription from said application adapter for said new object definition (see Kaipa, paragraph 37; wherein "subscription from" is interpreted as receiving the new object definition).

As to claim 37, Mitchell as modified by Kaipa, discloses:

further comprising instructions for forwarding said object in response to said subscription (see Kaipa, paragraph 39, "The resultant business object definition (in this case, preferably a Java object) is forwarded to the connector 310 for use in runtime object conversion").

As to claim 38, Mitchell as modified by Kaipa, discloses:

wherein said collaboration code comprise dynamically generated business object newly discovered during runtime (see Mitchell, paragraph 47), and provides an indication as to how the business object was processed (see Kaipa, paragraphs 18 and 35).

As to claim 39, Mitchell as modified by Kaipa, discloses:

Art Unit: 2166

wherein the processor for receiving the object and the collaboration code and for determining the object definition for said object based on said collaboration code, and the collaboration code for determining whether the object conforms to the known business object definition, comprise a reverse object discovery agent means (see Mitchell, paragraph 22).

As to claim 40, Mitchell as modified by Kaipa, discloses:

wherein determining the business object definition for said object without pre-defined business object definitions comprises:

determining a mapping information by determining how a plurality of business objects was merged to create the received object (see Kaipa, paragraphs 18 and 35);

creating the business object definition based on the determined mapping information (see Kaipa, paragraph 39);

sending the created business object definition to an adapter (see Kaipa, paragraph 37, lines 2-6); and

subscribing to the new business object definition (see Kaipa, paragraph 39, "The resultant business object definition (in this case, preferably a Java object) is forwarded to the connector 310 for use in runtime object conversion"; wherein "subscribing to" is interpreted as receiving the new object definition).

As to claim 41, Mitchell as modified by Kaipa, discloses:

wherein said receiving the object and the collaboration code is performed before said

Art Unit: 2166

determining the business object definition (see Mitchell, paragraphs 33 and 35), and wherein said storing the business object definition is performed before said determining the business object definition (see Mitchell, paragraphs 22 and 25; "wherein storing before..." is interpreted as using pre-defined definitions).

Note: There are several recitations of storing the business object definition throughout the claims. The examiner is unable to discern exactly what is meant because throughout the specification, various meanings are given to this phrase. For example in paragraphs 5, 65, and 69 of the publication (US 20050216282 A1).

As to claim 20, Lai discloses:

a broker comprising a collaboration that receives a first business object and a first business object definition (see paragraphs 45, 56, 63) that generates a second business object without a predetermined business object definition (see paragraph 45), and that generates a collaboration code (see paragraph 240), (Note: who is receiving and generating? this is not clear at all. What is the purpose of receiving a definition and then not using it to generate an object without the definition?), and a processor serving to execute a reverse object discovery agent that receives the first business object definition, the second business object and a collaboration code from the broker and that defines a newly discovered business object definition for the second business object as a second business object definition during runtime as a based upon said collaboration code (see paragraphs 41, 45, 56, 63, 414),.

Art Unit: 2166

wherein the reverse object discovery agent determines a structure of the second business object definition by converting a structure of the second business object directly, by examining an instruction that creates the second business object (see paragraphs 41, 45, 56, 63, 414).

However, Lai does not explicitly disclose:

wherein said code reverse object discovery agent determines said second business object definition for said second business object without pre-defined business object definitions, if the reverse object discovery agent determines that the second business object does not conform to a known business object definition

Kaipa discloses:

wherein said code reverse object discovery agent determines said second business object definition for said second business object without pre-defined business object definitions, if the reverse object discovery agent determines that the second business object does not conform to a known business object definition

(see paragraphs 34, lines 4-8 and 38, lines 3-5; wherein collaboration code is interpreted as procedure(s) to manipulate the data/ object).

It would have been obvious, at the time the invention was made, to have modified the teaching of Lai by the teaching of Kaipa to provide a better way to carry out the conversion of XML schema to Java and other object definitions (see Kaipa, paragraph 9).

Art Unit: 2166

Note: the examiner is unable to ascertain what is generating the collaboration code, the ODA or the broker. Also, it is unclear as to what is meant by the collaboration code. Is the code actual instructions or, for example, a set of symbols. The phrase :”a collaboration code” is recited throughout and leads one to interpret this as a set of symbols. However, if this deals with business integration collaborations, then the claims should be amended so that one will interpret "collaboration code" as a set of instructions.

As to claim 21, Lai as modified by Kaipa, discloses:

agent determines said second business object definition for said second business object by requesting the collaboration code and the first business object definition from the broker, and discovers a mapping information that was used by tile broker to generate the second business object, if the reverse object discovery agent determines that the second business object does not conform to the known business object definition, and

the reverse object discovery agent creates the second business object definition based upon the discovered mapping information (see Kaipa, paragraphs 18, 35, and 39).

As to claim 22, Lai as modified by Kaipa, discloses:

wherein said collaboration code determines how said second object is derived from said first object (see Lai, paragraph 56; wherein, The ebXML architecture also allows business process collaboration using a business process specification shared between

Art Unit: 2166

the business applications).

As to claim 23, Lai as modified by Kaipa, discloses:

wherein said collaboration code provides all indication as to how the first business object was processed to obtain the second business object (see Lai, paragraph 56; wherein, The ebXML architecture also allows business process collaboration using a business process specification shared between the business applications.).

As to claim 24, Lai as modified by Kaipa, discloses:

an application adapter that receives said second object and said second object definition from said reverse object discovery agent (see Kaipa, paragraph 37, lines 2-6).

Response to Arguments

5. Applicant's arguments with respect to claims 20-24 have been considered but are moot in view of the new ground(s) of rejection.

6. Applicant's arguments filed 7-9-2010 have been fully considered but they are not persuasive.

Applicant's arguments that the combination of Mitchell and Kaipa do not disclose all of the limitations in claims 1 and 39 are acknowledged but are not deemed persuasive.

As to claim 1, Mitchell discloses *receiving an object and a collaboration code* (see paragraphs 33 and 35; wherein collaboration code is interpreted as procedure(s) to manipulate the data/ object. The limitation is interpreted as such based on the claim language used. If the applicant/ attorney wishes to have the limitation interpreted in a different light, he/she should amend accordingly.) *determining a business object definition for said object based upon said collaboration code* (see paragraphs 34, lines 4-8 and 38, lines 3-5; wherein collaboration code is interpreted as procedure(s) to manipulate the data/ object; and, wherein the middleware system automatically creates a Meta Data Definition Object (MDDO) from the discoverable type definitions of the imported object. Also, The MDDO 109 may be a meta object instance of the class structure contained in the meta interfaces of the imported type.).

However, Mitchell does not explicitly disclose *storing said business object definition*,

Art Unit: 2166

wherein said collaboration code determines said business object definition for said object without pre-defined business object definitions, if the object does not conform to a known business object definition. Kaipa discloses this at paragraph 39, lines 1-3 and 7-10; wherein the definition stored in a business object (as in applicant's publication, paragraph 65) is interpreted as unknown because the schema is xml and the business object definition created is java.

As for the rationale that the applicant/ attorney feels that the examiner is missing, it would have been obvious, at the time the invention was made, to have modified the teaching of Mitchell by the teaching of Kaipa to provide a better way to carry out the conversion of XML schema to Java and other object definitions (see Kaipa, paragraph 9).

As to claim 39, Mitchell as modified by Kaipa, discloses *wherein the processor for receiving the object and the collaboration code and for determining the object definition for said object based on said collaboration code, and the collaboration code for determining whether the object conforms to the known business object definition, comprise a reverse object discovery agent means* (see Mitchell, paragraph 22). Here Mitchell discloses an object with meta data that may be discovered or reverse engineered; a middleware framework in which the aggregation may occur; importation of the object into the middleware framework; creation of a middleware Meta Data Definition Object (MDDO); creation of a runtime proxy based on the MDDO; runtime instantiation of aggregated portions of proxy as needed; and automated runtime

Art Unit: 2166

transformation of object interfaces. As for applicant's processor (which is inherently apparent), this can be found in paragraph 27 of Kaipa.

Because Mitchell, Kaipa, and the newly recited Lai disclose all limitations of the claimed invention, none of the claims are considered allowable.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Johnese Johnson whose telephone number is 571-270-1097. The examiner can normally be reached on 4/5/9.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hosain Alam can be reached on 571-272-3978. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/J. J./

Examiner, Art Unit 2166

September 28, 2010

/Hosain T Alam/

Supervisory Patent Examiner, Art Unit 2166